Terms used **BDD label leaves sifting nodes order top down** Found **2** of **158,639**

Sort results by: relevance
Display results: expanded form
Save results to a Binder
Search Tips
Open results in a new window

Try an Advanced Search
Try this search in The ACM Guide

Results 1 - 2 of 2

Relevance scale

1 Who are the variables in your neighborhood

Shipra Panda, Fabio Somenzi

December 1995 **Proceedings of the 1995 IEEE/ACM international conference on Computer-aided design**

Full text available: pdf(198.47 KB) Publisher Site

Additional Information: full citation, abstract, references, citings, index terms

Dynamic reordering techniques have had considerable success in reducing the impact of the initial variable order on the size of decision diagrams. Sifting, in particular, has emerged as a very good compromise between low CPU time requirements and high quality of results. Sifting, however, has the absolute position of a variable as the primary objective, and only considers the relative positions of groups of variables indirectly. In this paper we propose an extension to sifting that may move grou ...

2 Tutorial: Compiling concurrent languages for sequential processors

Stephen A. Edwards

April 2003 **ACM Transactions on Design Automation of Electronic Systems (TODAES)**, Volume 8 Issue 2

Full text available: pdf(771.65 KB)

Additional Information: full citation, abstract, references, citings, index terms, review

Embedded systems often include a traditional processor capable of executing sequential code, but both control and data-dominated tasks are often more naturally expressed using one of the many domain-specific concurrent specification languages. This article surveys a variety of techniques for translating these concurrent specifications into sequential code. The techniques address compiling a wide variety of languages, ranging from dataflow to Petri nets. Each uses a different method, to some degr ...

Keywords: Compilation, Esterel, Lustre, Petri nets, Verilog, code generation, communication, concurrency, dataflow, discrete-event, partial evaluation, sequential

Results 1 - 2 of 2

Terms used **BDD leaves sifting nodes order** Found **18** of **158,639**

Sort results by relevance
Display results expanded form
Save results to a Binder
Search Tips
Open results in a new window

Try an Advanced Search
Try this search in The ACM Guide

Results 1 - 18 of 18

Relevance scale

1 Symmetry detection and dynamic variable ordering of decision diagrams
Shipra Panda, Fabio Somenzi, Bernard F. Plessier
November 1994 **Proceedings of the 1994 IEEE/ACM international conference on Computer-aided design**
Full text available: pdf(515.23 KB) Additional Information: full citation, abstract, references, citings, index terms

Knowing that some variables are symmetric in a function has numerous applications; in particular, it can help produce better variable orders for Binary Decision Diagrams (BDDs) and related data structures (e.g., Algebraic Decision Diagrams). It has been conjectured that there always exists an optimum order for a BBD wherein symmetric variables are contiguous. We propose a new algorithm for the detection of symmetries, based on dynamic reordering, and we study its interaction with the reorde ...

2 Lazy group sifting for efficient symbolic state traversal of FSMs
Hiroyuki Higuchi, Fabio Somenzi
November 1999 **Proceedings of the 1999 IEEE/ACM international conference on Computer-aided design**
Full text available: pdf(108.64 KB) Additional Information: full citation, abstract, references, citings, index terms

This paper proposes lazy group sifting for dynamic variable re-ordering during state traversal. The proposed method relaxes the idea of pairwise grouping of present state variables and their corresponding next state variables. This is done to produce better variable orderings during image computation without causing BDD size blowup in the substitution of next state variables with present state variables at the end of image computation. Experimental results show that our approach is more rob ...

3 Functional simulation using binary decision diagrams
Christoph Scholl, Rolf Drechsler, Bernd Becker
November 1997 **Proceedings of the 1997 IEEE/ACM international conference on Computer-aided design**
Full text available: pdf(219.04 KB) Publisher Site Additional Information: full citation, abstract, references, citings, index terms

In many verification techniques fast functional evaluation of a Boolean network is needed. We investigate the idea of using Binary Decision Diagrams (BDDs) for functional simulation. The area-time trade-off that results from different minimization techniques of the BDD is discussed. We propose new minimization methods based on dynamic reordering that allow

smaller representations with (nearly) no runtime penalty.

Keywords: Functional simulation, Binary Decision Diagrams

4 Expressions for Boolean functions: Minimization of the expected path length in BDDs based on local changes

Rüdiger Ebendt, Wolfgang Günther, Rolf Drechsler

January 2004 **Proceedings of the 2004 conference on Asia South Pacific design automation: electronic design and solution fair ASP-DAC '04 , Proceedings of the 2004 conference on Asia South Pacific design automation: electronic design and solution fair ASP-DAC '04**

Full text available: pdf(126.17 KB) Publisher Site Additional Information: full citation, abstract, references

In many verification tools methods for functional simulation based on reduced ordered Binary Decision Diagrams (BDDs) are used. The evaluation time for a BDD can be crucial and is measured by the expected path length of the BDD.In this paper a new technique for BDD minimization with respect to the expected path length is suggested to reduce evaluation time. It is based on sifting and, unlike previous approaches, performs variable swaps with the same time complexity as the original sifting algori ...

5 Who are the variables in your neighborhood

Shipra Panda, Fabio Somenzi

December 1995 **Proceedings of the 1995 IEEE/ACM international conference on Computer-aided design**

Full text available: pdf(198.47 KB) Publisher Site Additional Information: full citation, abstract, references, citings, index terms

Dynamic reordering techniques have had considerable success in reducing the impact of the initial variable order on the size of decision diagrams. Sifting, in particular, has emerged as a very good compromise between low CPU time requirements and high quality of results. Sifting, however, has the absolute position of a variable as the primary objective, and only considers the relative positions of groups of variables indirectly. In this paper we propose an extension to sifting that may move grou ...

6 Don't care-based BDD minimization for embedded software

Youpyo Hong, Peter A. Beerel, Luciano Lavagno, Ellen M. Sentovich

May 1998 **Proceedings of the 35th annual conference on Design automation**

Full text available: pdf(167.37 KB) Publisher Site Additional Information: full citation, abstract, references, citings, index terms

This paper explores the use of don't cares in software synthesis for embedded systems. Embedded systems have extremely tight real-time and code/data size constraints, that make expensive optimizations desirable. We propose applying BDD minimization techniques in the presence of a don't care set to synthesize code for extended Finite State Machines from a BDD-based representation of the FSM transition function. The don't care set can be derived from local analysis (such as unused sta ...

Keywords: MPEG4, codec, design automatian, flip-flops, level converters, low power, placement, synthesis, voltage scaling

7 Synthesis of wiring signature-invariant equivalence class circuit mutants and applications to benchmarking

D. Ghosh, N. Kapur, J. Harlow, F. Brglez
February 1998 **Proceedings of the conference on Design, automation and test in Europe**

Full text available: pdf(413.83 KB) Publisher Site Additional Information: full citation, abstract, references, citings, index terms

This paper formalizes the synthesis process of wiring signature-invariant (WSI) combinational circuit mutants. The signature \sigma_0 is defined by a reference circuit \eta_0, which itself is modeled as a canonical form of a directed bipartite graph. A wiring perturbation \gamma induces a perturbed reference circuit \eta_{\gamma}. A number of mutant circuits \eta_{\gamma_i} can be resynthesized from the perturbed circuit \eta_\gamma. The mutants of interest are the ones that belong to the wiring ...

Keywords: signature-invariance, equivalence class, circuit mutants, benchmarking

8 Speeding up symbolic model checking by accelerating dynamic variable reordering
Christoph Meinel, Christian Stangier
March 2000 **Proceedings of the 10th Great Lakes symposium on VLSI**

Full text available: pdf(500.80 KB) Additional Information: full citation, abstract, references, citings, index terms

Symbolic Model checking is a widely used technique in sequential verification. As the size of the OBDDs and also the computation time depends on the order of the input variables, the verification may only succeed if a well suited variable order is chosen. Since the characteristics of the represented functions are changing, the variable order has to be adapted dynamically. Unfortunately, dynamic reordering strategies are often very time consuming and sometimes do not provide any improvement of ...

9 Synthesis of software programs for embedded control application
Massimiliano Chiodo, Paolo Guisto, Attila Jurecska, Luciano Lavagno, Ellen Sentovich, Harry Hsieh, Kei Suzuki, Alberto Sangiovanni-Vincentelli
January 1995 **Proceedings of the 32nd ACM/IEEE conference on Design automation**

Full text available: pdf(127.40 KB) Additional Information: full citation, references, citings, index terms

10 Symbolic pointer analysis revisited
Jianwen Zhu, Silvian Calman
June 2004 **ACM SIGPLAN Notices , Proceedings of the ACM SIGPLAN 2004 conference on Programming language design and implementation**, Volume 39 Issue 6

Full text available: pdf(256.99 KB) Additional Information: full citation, abstract, references, citings, index terms

Pointer analysis is a critical problem in optimizing compiler, parallelizing compiler, software engineering and most recently, hardware synthesis. While recent efforts have suggested symbolic method, which uses Bryant's Binary Decision Diagram as an alternative to capture the point-to relation, no speed advantage has been demonstrated for context-insensitive analysis, and results for context-sensitive analysis are only preliminary.In this paper, we refine the concept of symbolic transfer functio ...

Keywords: binary decision diagrams, call graph construction, pointer analysis

11 The Compositional Far Side of Image Computation
Chao Wang, Gary D. Hachtel, Fabio Somenzi
November 2003 **Proceedings of the 2003 IEEE/ACM international conference on**

Computer-aided design

Full text available: pdf(205.38 KB) Additional Information: full citation, abstract, index terms

Symbolic image computation is the most fundamental computationin BDD-based sequential system optimization and formal verification.In this paper, we explore the use of over-approximationand BDD minimization with donýt cares during image computation.Our new method, based on the partitioned representation ofthe transition relation, consists of three phases: First, the model istreated as a set of loosely coupled components, and over-approximateimages are computed to minimize the transition relation ...

Keywords: Model checking, image computation, Binary Decision Diagrams, Symbolic

12 Free MDD-based software optimization techniques for embedded systems

Chunghee Kim, Luciano Lavagno, Alberto Sangiovanni-Vincentelli

January 2000 **Proceedings of the conference on Design, automation and test in Europe**

Full text available: pdf(212.34 KB) Publisher Site Additional Information: full citation, references, citings, index terms

13 Tutorial: Compiling concurrent languages for sequential processors

Stephen A. Edwards

April 2003 **ACM Transactions on Design Automation of Electronic Systems (TODAES)**, Volume 8 Issue 2

Full text available: pdf(771.65 KB) Additional Information: full citation, abstract, references, citings, index terms, review

Embedded systems often include a traditional processor capable of executing sequential code, but both control and data-dominated tasks are often more naturally expressed using one of the many domain-specific concurrent specification languages. This article surveys a variety of techniques for translating these concurrent specifications into sequential code. The techniques address compiling a wide variety of languages, ranging from dataflow to Petri nets. Each uses a different method, to some degr ...

Keywords: Compilation, Esterel, Lustre, Petri nets, Verilog, code generation, communication, concurrency, dataflow, discrete-event, partial evaluation, sequential

14 Session 10C: flexibility in logic synthesis: Generalized symmetries in boolean functions

Victor N. Kravets, Karem A. Sakallah

November 2000 **Proceedings of the 2000 IEEE/ACM international conference on Computer-aided design**

Full text available: pdf(134.37 KB) Additional Information: full citation, abstract, references, citings

In this paper we take a fresh look at the notion of symmetries in Boolean functions. Our studies are motivated by the fact that the classical characterization of symmetries based on invariance under variable swaps is a special case of a more general invariance based on unrestricted variable permutations. We propose a generalization of classical symmetry that allows for the simultaneous swap of ordered and unordered groups of variables, and show that it captures more of a function's invariant per ...

15 Formal verification of superscale microprocessors with multicycle functional units, exception, and branch prediction

Miroslav N. Velev, Randal E. Bryant

June 2000 **Proceedings of the 37th conference on Design automation**

Full text available: pdf(69.74 KB) Additional Information: full citation, abstract, references, citings, index terms

We extend the Burch and Dill flushing technique [6] for formal verification of microprocessors to be applicable to designs where the functional units and memories have multicycle and possibly arbitrary latency. We also show ways to incorporate exceptions and branch prediction by exploiting the properties of the logic of Positive Equality with Uninterpreted Functions [4][5]. We study the modeling of the above features in different versions of dual-issue superscalar processors.

16 Boolean satisfiability with transitivity constraints

Randal E. Bryant, Miroslav N. Velev

October 2002 **ACM Transactions on Computational Logic (TOCL)**, Volume 3 Issue 4

Full text available: pdf(284.90 KB) Additional Information: full citation, abstract, references, citings, index terms

We consider a variant of the Boolean satisfiability problem where a subset ϵ of the propositional variables appearing in formula *F*sat encode a symmetric, transitive, binary relation over *N* elements. Each of these *relational* variables, *ei,j*, for $1 \leqslant i < j \leqslant N$, expresses whether or not the relation holds between elements *i* and *j*. The task is to either find a satisfying assignment to *F*< ...

Keywords: Boolean satisfiability, decision procedures, formal verification

17 Increasing efficiency of symbolic model checking by accelerating dynamic variable reordering

Christoph Meinel, Christian Stangier

January 1999 **Proceedings of the conference on Design, automation and test in Europe**

Full text available: pdf(23.26 KB) Additional Information: full citation, index terms

18 Improving Ariadneýs Bundle by Following Multiple Threads in Abstraction Refinement

Chao Wang, Bing Li, HoonSang Jin, Gary D. Hachtel, Fabio Somenzi

November 2003 **Proceedings of the 2003 IEEE/ACM international conference on Computer-aided design**

Full text available: pdf(296.88 KB) Additional Information: full citation, abstract, index terms

We propose an abstraction refinement method for invariant checking,which is based on the simultaneous analysis of all abstractcounter examples of shortest length in the current abstraction. Thealgorithm is focused on an improved Ariadne's Bundle of SORs(Synchronous Onion Rings) of the abstract model; the transitionsthrough these SORs contain all shortest ACEs (Abstract CounterExamples) and no other ACEs. The SORs are exploited in twodistinct ways to provide global guidance to the abstraction ref ...

Results 1 - 18 of 18